

Orientaatiotehtävä

Tutustu johdantolukuun "A Brief Introduction to C++" kirjasta

Andrist, B., Sehr, V., C++ High Performance - Second Edition. Packt, 2020.

Johdantoluku (Chapter 1: A brief introduction to C++) on luettavissa ilmaiseksi [täältä](#) ("Click to preview"). Jos et ole kovin kokenut C++ -ohjelmoija, saattavat useat johdannon aiheet ja koodiesimerkit vaikuttaa heprealta. Älä kuitenkaan anna periksi, vaan lue koko teksti, ja pyri muodostamaan jonkinlainen yleiskuva käsiteltävistä asioista.

Pyri siihen, että osaisit lopulta vastata jollakin tasolla seuraaviin kysymyksiin seuraavaan luentoon mennessä (kirjaa esim. muutama ranskalainen viiva joka kohdasta):

1. Mitä "abstraktioilla" tarkoitetaan ohjelmointikielten yhteydessä?
2. Mitä tarkoitetaan käsitteellä "zero-cost abstractions"? Millainen C++ on tässä suhteessa verrattuna muihin ohjelmointikieliin?
3. Mitä tarkoittaa "zero-overhead"-periaate? Mitkä C++:n ominaisuudet EIVÄT noudata tätä periaatetta?
4. Mitkä C++:n ominaisuudet edesauttavat toimintavarman ("robust") koodin kirjoittamista?
5. Miten Java-ohjelman kääntäminen eroaa C++ -ohjelman kääntämisestä?
6. Miten dynaamisesti varatun muistin (olioiden) vapautus eroaa Javassa ja C++ :ssa?
7. Millä tavoilla C++ tarjoaa enemmän vaihtoehtoja olioiden allokointiin kuin Java?
8. Miksi käytettävät oliot olisi hyvä sijoittaa lähekkäin muistissa?
9. Mitä C++:n tarjoamat "value semantics" ja "reference semantics" tarkoittavat?
10. Yritä selvittää, mitä tekstissä esiintyneet `std::shared_ptr` ja `std::weak_ptr` ovat.
11. Mitä "const correctness" -konsepti tarkoittaa?
12. Useassa koodiesimerkissä käytetään "auto"-avainsanaa. Mitä se mahtaa tarkoittaa?
13. Mitä tarkoittaa, että olioiden tuhoaminen on C++:ssa determinististä ("deterministic destruction"). Miksi se on hyvä asia?
14. Miten null-referenssien käsittely poikkeaa Javassa ja C++:ssa?
15. Mitkä ovat tärkeimmät artikkelissa mainitut C++:n puutteet? Mitä mieltä itse olet?